

Estratégia de Paralelização Computacional Aplicada em Técnicas de Traçado de Raios 3D para Modelagem de Canal Rádio-Móvel

A. M. Cavalcante, L. V. de Souza, M. J. de Sousa, J. C. W. A. Costa, C. R. L. Francês, *et al.*
UFPA – DEEC – LEA-- Rua Augusto Corrêa, No. 01, Guamá - Caixa Postal 8619 - CEP 66.075-900 – Belém – Pará

Resumo — Este artigo propõe uma estratégia de paralelização computacional aplicada em modelos de propagação baseados nas técnicas de traçado de raios 3D (*Ray-tracing 3D*). A estratégia abordada baseia-se no conceito de independência dos raios, que permite uma divisão uniforme das tarefas através da distribuição equitativa dos raios entre os nós formadores do computador paralelo (*cluster*).

Palavras-chaves — Computação paralela, *cluster* de computadores, técnica de traçado de raios, canal de propagação.

I. INTRODUÇÃO

A evolução ocorrida no *hardware* da década de 80 (aumento da capacidade de processamento e redução dos custos dos processadores) acarretou na popularização e difusão dos computadores pessoais e estações de trabalho. Com igual avanço ocorrendo nas técnicas de comunicação entre computadores (redes com 10 Mbps, 100 Mbps e velocidades superiores), estava criado o ambiente propício para o compartilhamento de recursos como processadores, dispositivos de impressão e demais periféricos. Entretanto, este compartilhamento de recursos possui problemas tais como a dificuldade de concepção de um controle distribuído ou central e a gerência do fluxo de mensagens entre todos os processos envolvidos.

Ferramentas voltadas para processamento paralelo como MPI (*Message Passing Interface*), PVM (*Parallel Virtual Machine*), ConcertC (um extensão da linguagem C para programação paralela e distribuída) e Voyager (programação distribuída em Java) tornam possíveis a utilização de um conjunto de máquinas rodando um único processo conjuntamente [1]-[2].

A redução do tempo gasto nas simulações permite um maior ganho de produtividade e a possibilidade de utilização de soluções numéricas mais precisas. Adicionalmente, o uso de técnicas de aceleração computacional possibilita que problemas mais complexos sejam pesquisados e tratados, uma vez que tal complexidade implica em alto processamento computacional.

Um exemplo apropriado de aplicação está na predição de propagação eletromagnética utilizando a técnica de traçado de raios. A técnica de modelagem por traçado de raios em 3D

para ambientes *indoor* e *outdoor* possui grande utilização e eficácia [3]-[5]. Entretanto, um limitante intrínseco dessa técnica é que o esforço computacional empregado é proporcional à complexidade e ao número de faces do cenário.

Com a utilização de técnicas de melhoria de desempenho computacional, é possível otimizar a simulação de propagação de sinais de rádio-frequência (RF) em ambientes móveis, diminuindo o tempo total da simulação e tornando possível a análise de outros ambientes *wireless* mais complexos.

Este artigo está dividido da seguinte forma: na seção II são apresentadas as características básicas do processamento paralelo; a seção III mostra detalhes das técnicas de modelagem por traçado de raios e a sua relação com a computação paralela; a seção IV discute as estratégias de paralelização propostas, e a seção V apresenta as conclusões.

II. PROCESSAMENTO PARALELO

A paralelização é uma estratégia utilizada em computação para se obter resultados mais rápidos de grandes e complexas tarefas. Uma grande tarefa pode ser executada serialmente ou paralelamente, dividida em partes da seguinte forma [1]:

1. Identificam-se tarefas menores no interior da tarefa maior a ser paralelizada;
2. Distribuem-se as pequenas tarefas por entre vários “trabalhadores”, que irão executá-la simultaneamente;
3. Coordenam-se os “trabalhadores”.

Para se decompor um programa em pequenas tarefas a serem executadas em paralelo, primeiro é necessário distinguir a decomposição funcional da decomposição de domínio. Na primeira, o problema é decomposto em diferentes tarefas que serão distribuídas por entre múltiplos processadores para execução simultânea. Essa técnica é adequada a um programa dinâmico e modular. Cada tarefa será um programa diferente. Na decomposição de domínio, os dados são decompostos em grupos que serão distribuídos por entre múltiplos processadores que executarão, simultaneamente, um mesmo programa [1]-[2].

Seja qual for a forma de decomposição escolhida, deve-se garantir o balanceamento de carga. A distribuição das tarefas por entre os processadores deverá ser sempre de maneira tal que o tempo da execução das partes seja o mais uniforme possível. Se essa exigência não for satisfeita, o desempenho

* L. V de Souza, lvsouza@nautilus.com.br, A. M. Cavalcante, amc@ufpa.br, M. J. de Souza, mjs@ufpa.br, J. C. W. A. Costa, jweyl@ufpa.br, C. R. L. Francês, rfrances@ufpa.br, G. P. S. Cavalcante, gervasio@ufpa.br, Tel +55-91-2111302, Fax +55-91-2111634.

Este trabalho foi parcialmente financiado pelo CNPq e pela Ericsson - Projeto UFA01.

do programa paralelo será menor, pois o tempo total da aplicação será determinado pelo processo mais demorado.

Um *cluster* é um sistema formado por dois ou mais computadores, os quais trabalham em conjunto para executar aplicações ou realizar determinadas tarefas. O *cluster* Beowulf foi idealizado pelos seus desenvolvedores com o objetivo de suprir a crescente e elevada demanda por capacidade de processamento em diversas áreas científicas, aliando a isto o uso sistemas computacionais poderosos e economicamente viáveis [1].

Uma característica chave de um *cluster* Beowulf, é o *software* utilizado, que é de elevado desempenho e gratuito na maioria de suas ferramentas, como exemplo os sistemas operacionais GNU/Linux e FreeBSD, sobre os quais estão instaladas as diversas ferramentas que viabilizam o processamento paralelo, como é o caso do MPI (*Message Passing Interface*) e PVM (*Parallel Virtual Machine*) [1]-[2].

A proposta de paralelização apresentada neste trabalho é baseada nas prerrogativas dos *clusters* Beowulf, tendo como respaldo outros trabalhos realizados pelo grupo de pesquisa utilizando computação paralela, tais como a síntese de grades de Bragg em fibra [6] e aplicações de Diferenças Finitas no Domínio do Tempo [7].

III. TÉCNICA DE TRAÇADO DE RAIOS

Existem basicamente duas abordagens para se implementar uma técnica de traçado de raios aplicada à modelagem da propagação de ondas eletromagnéticas: uma baseada na Teoria das Imagens (IT – *Image Theory*) [8],[9] e outra na técnica SBR (*Shooting-and-Bouncing Ray*), também conhecida como “Força-Bruta” [3]-[5], [10].

Na IT, são determinados os pontos imagem em relação às faces do cenário que definirão percursos envolvendo reflexões entre a antena transmissora e a receptora. Adicionalmente, pode-se incorporar algoritmos de busca por pontos de difração para se definir percursos difrativos. Mecanismos de transmissão, que representam uma importante contribuição principalmente em ambientes *indoor*, geralmente não são incorporados a IT, por apresentarem algoritmos de busca por pontos de transmissão relativamente complexos. A aplicação da IT, entretanto, é muito limitada pela dimensão e a complexidade do cenário, pois a maior parte dos recursos computacionais são gastos basicamente para a geração da árvore de imagens (estrutura de dados contendo todos os pontos imagens em relação às faces do cenário que definirão potenciais percursos envolvendo reflexões entre o transmissor e o receptor), sendo que o tamanho desta estrutura cresce de forma exponencial com o aumento do número de faces do cenário, necessitando assim, de grandes recursos principalmente de memória.

A técnica SBR por outro lado, consiste basicamente no lançamento de raios a partir da antena transmissora e no mapeamento do percurso percorrido por cada raio e seus descendentes, contabilizando assim, todos os mecanismos (reflexão, transmissão e difração) e as combinações dos mesmos. Nesse contexto, o principal esforço computacional utilizado por essa técnica é gasto nos procedimentos de testes

de somreamento (interseção raio-face) do cenário, o que dependerá do número de faces do cenário.

Desse modo, técnicas de traçado de raios podem exigir muito em termos de recursos computacionais, especialmente quando as dimensões do cenário e principalmente o número de faces de cenário crescem. Para tornar a execução do programa que implementa uma técnica de traçado de raios mais rápida, sem comprometer a qualidade dos resultados gerados, são utilizadas técnicas denominadas de Técnicas de Aceleração. A primeira abordagem para se reduzir o tempo computacional gasto na execução do programa é de se otimizar os procedimentos de testes de somreamento. A idéia básica dessa otimização consiste no fato de que não é necessário testar a interseção de um raio com todas as faces do cenário, e sim, com apenas um conjunto delas. De forma geral, tais técnicas consistem em dividir o cenário em regiões, segundo algum critério, reduzindo com isso, o número de testes de somreamento realizados. As principais técnicas de aceleração usadas no particionamento de cenário são [8]:

- *BSP (Binary Space Partition)*: Essa técnica particiona o cenário gerando uma estrutura de dados chamada de “árvore BSP”. Essa árvore é função apenas da distribuição geométrica das faces do cenário sendo, portanto, independente da localização da antena transmissora e dos pontos de recepção. A filosofia dessa técnica é que, uma vez construída a árvore BSP (e apenas uma vez para um dado cenário), o tempo de execução restante seja otimizado, através da consulta à árvore a cada teste de somreamento;
- *SVP (Space Volumetric Partition)*: Essa técnica é semelhante a BSP, no sentido de que o cenário é também particionado em regiões, independente do posicionamento da antena transmissora e dos pontos de recepção. Nessa técnica, o cenário é subdividido em volumes, de forma também independente da localização das faces do cenário (ao contrário da técnica BSP), gerando o que se denomina “matriz SVP”. Da mesma forma que na técnica anterior, a matriz é gerada uma única vez para o cenário e o objetivo é reduzir o número de faces a serem testadas na realização de testes de somreamento, através da consulta à matriz SVP.
- *Algoritmo Angular Z-buffer*: Esta técnica se diferencia das anteriores no aspecto de que a divisão do espaço é dependente do posicionamento da fonte dos raios (e apenas desse parâmetro). Nessa técnica, o espaço angular do cenário em volta da fonte é dividido em regiões angulares, gerando matrizes denominadas de AZB. Através da consulta às matrizes AZB quando da realização de testes de somreamento, o número de faces a serem testadas será reduzido.

A outra abordagem referente a Técnicas de Aceleração é a de paralelização do código do programa que implementa os algoritmos de traçado de raios, ou seja, distribuir as tarefas de processamento entre vários processadores. Na IT, esta abordagem não é trivial, pois a estrutura de dados adotada (árvore de imagens) é totalmente concatenada, dificultando a

divisão das tarefas e o balanceamento de carga entre os processadores. A técnica SBR por sua concepção já é intrinsecamente paralelizável, pois os raios lançados pela antena transmissora são independentes. Podendo ser aplicável, portanto, diretamente no contexto de programação paralela. A tarefa que seria realizada por um único processador (lançamento e mapeamento de n raios e seus descendentes, juntamente com seus respectivos testes de sombreamento) agora será dividida entre os processadores que compõem o *cluster*. Devido a esta independência dos raios, o programa apresentará uma granulosidade grossa (muito processamento e pouca comunicação entre os processos), resultando assim, em um *speed-up* (relação entre o tempo de execução serial e o tempo de execução paralela) aproximadamente linear, e uma eficiência (relação entre o *speed-up* e o número de processadores) de paralelização de quase 100% [1]. Nesse contexto, o tempo de processamento paralelo em uma técnica SBR será inversamente proporcional ao número de processadores utilizados no *cluster*, sem saturação, possibilitando assim, que o tempo de execução seja reduzido o tão quanto seja possível incorporar novos processadores no *cluster*.

IV. ESTRATÉGIAS DE PARALELIZAÇÃO

Para o algoritmo de traçados de raios SBR, o elemento fundamental e ativo é o raio. Um raio consiste em um caminho possível (exato ou aproximado) de propagação dentro de um ambiente de propagação simulado. Cada raio é criado de forma independente e também interage independentemente com o meio até que sua energia se torne pequena a ponto de poder ser ignorada, sendo o raio, portanto, descartado. Sendo todo o esforço computacional associado aos objetos “raio”, e sendo esses tão independentes uns dos outros, pode-se imaginar uma estratégia de paralelização na qual os raios são simplesmente distribuídos entre os vários nós processadores que formam o computador paralelo. Apenas o cenário, antenas transmissora e receptora seriam compartilhados, porém estes objetos poderiam ser replicados e suas cópias mantidas fisicamente na memória principal de cada nó do *cluster*.

Assim sendo, uma importante questão pode ser levantada: sendo os raios absolutamente independentes uns dos outros, não há necessidade de comunicação entre os nós processadores do *cluster* durante a etapa de simulação de lançamentos de raios; etapa esta que concentra praticamente todo o custo computacional do programa SBR (Fig. 1).

Sem a necessidade de comunicação, a eficiência de paralelização seria teoricamente sempre de 100%, independente da quantidade de processadores utilizados. Contudo, para essa implementação existem alguns problemas que precisam ser resolvidos para que seja realmente possível minimizar a comunicação entre os nós do *cluster*. Todos esses problemas estão associados a particularidades no processo de simulação dos raios, que podem comprometer o balanceamento da carga entre os nós processadores.

O processamento de um raio consiste na efetivação de repetitivos testes de interceptação deste com todas as faces que compõem o cenário. Essa tarefa não só demanda algum

custo computacional, como também gera novos raios. Quando um raio atinge uma superfície como, por exemplo, a parede de um prédio, o programa passa a considerar mais dois raios resultantes: o refletido e o refratado (Fig. 2).

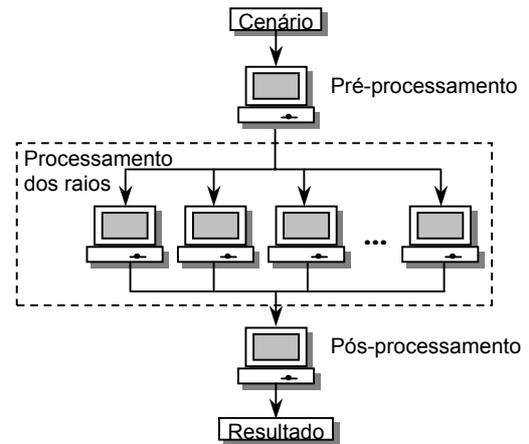


Fig. 1. Proposta de paralelização para o algoritmo SBR com a etapa de processamento intensivo (em destaque) constituída por nós isolados.

Porém, quando um raio atinge uma aresta, como uma borda (quina) resultante junção entre duas paredes, teoricamente deveria ocorrer a geração de infinitos raios devido à difração (Fig. 2). Na prática, o algoritmo gera uma quantidade finita de raios, tanto maior quanto mais acurada for a precisão imposta ao modelo. Portanto, a geração de novos raios é progressiva de crescimento exponencial. Porém, a partir do momento em que a energia dos raios descendentes atinge valores suficientemente pequenos, a taxa de “mortalidade” de raios começa a superar a taxa de “natalidade” e o tamanho da população de raios passa a reduzir ao invés de aumentar. O número máximo de raios depende fundamentalmente da complexidade do cenário. Assim, dois raios disparados em direções diferentes podem gerar um número diferente de raios derivados, já que normalmente os cenários não são homoganeamente complexos. Como a paralelização prescreve a divisão dos raios entre os nós processadores, e como pode ocorrer que os descendentes gerados por esses raios sejam em números diferentes, dependendo da complexidade do cenário para o qual eles são dirigidos, a consequência seria o desbalanceamento de carga entre os nós o que comprometeria a paralelização.

Para solucionar esse problema seria preciso conhecer antecipadamente qual seria o comportamento de cada raio gerado a partir da antena transmissora. Infelizmente esse comportamento só pode ser determinado através da simulação dos raios, que é justamente o problema a ser resolvido. Porém, seria possível realizar um balanceamento adaptativo do problema, dividindo continuamente os raios entre os nós processadores à medida que os raios são processados. Na prática esse “balanceamento adaptativo” exigiria intensa comunicação entre os nós, justamente o que se deseja evitar. Certamente a solução adequada para o problema é descobrir uma maneira de distribuir corretamente os raios iniciais conhecidos, oriundos da antena transmissora, entre os vários processadores do *cluster*.

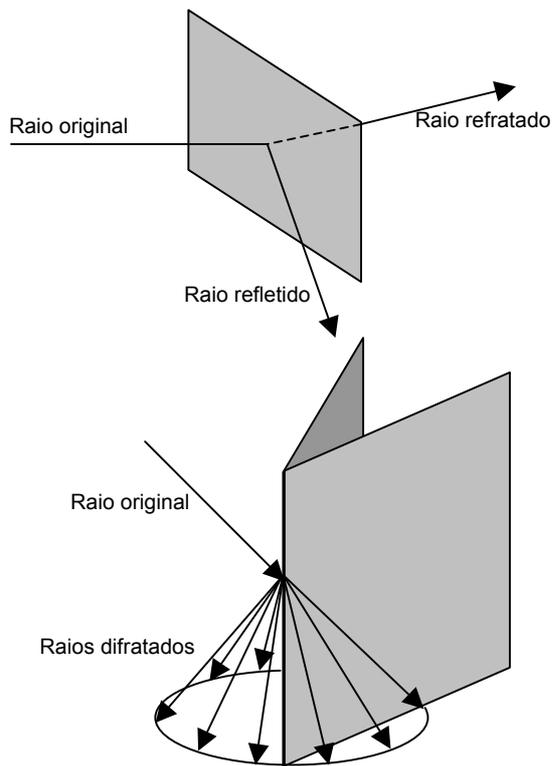


Fig. 2. Processos de geração de raios envolvidos na técnica de traçado de raios

- (a) dois raios através dos fenômenos da reflexão e refração.
- (b) inúmeros raios através do fenômeno da difração.

Uma das melhores formas de modelagem da fonte emissora de raios é baseada na subdivisão recursiva dos lados de um icosaedro, como mostrado na Fig. 3 [11]. A partir da projeção desses vértices coplanares em uma superfície esférica de raio unitário concêntrica ao icosaedro, obtém-se os raios iniciais emitidos. Uma forma simplista de divisão desses raios seria a divisão em partes do icosaedro entre os nós do *cluster*, ficando cada nó, responsável pelo processamento dos raios que surgirem em seu domínio. Porém, pode ocorrer que uma dessas partes esteja voltada para uma região do espaço repleta de obstáculos, enquanto outra esteja voltada para uma região absolutamente vazia. Nesse caso, o balanceamento de carga seria comprometido.

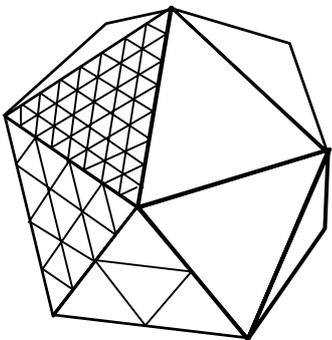


Fig. 3. Obtenção dos pontos dos raios iniciais através da subdivisão fractal (em triângulos) das faces do icosaedro [11].

A proposta defendida neste trabalho é formar o conjunto de raios de cada nó através de uma escolha aleatória efetuada sobre todos os raios que compõem a fonte transmissora. Dessa forma, cada região do icosaedro seria constituída de raios cujo processamento seria responsabilidade de nós processadores variados. A distribuição aleatória dos raios seria mais eficiente quanto maior for o número total de raios emitidos, exatamente o caso quando mais se justifica o uso da computação paralela. Através dessa técnica, a carga de processamento de um *cluster* homogêneo seria balanceada através da distribuição de um número igual de raios (aleatoriamente escolhidos) para cada nó. Para um *cluster* heterogêneo, o número de raios de cada nó deve ser proporcional a sua capacidade de processamento. Evidentemente, descobrir a capacidade de processamento de computadores é algo que se pode fazer antecipadamente, sendo até possível estimá-la com base em características de *hardware* e *software*.

Especificamente, o pré-processamento mostrado na Fig. 1 consistiria na separação dos raios iniciais para cada nó do *cluster*. Essa é a única diferença existente entre os processos de cada nó: quais os raios iniciais serão simulados dentre todos pertencentes ao icosaedro de transmissão; o que caracteriza a solução proposta neste artigo como SPMD (*Single Program Multiple Data*) [1].

As diferentes listagens de raios seriam distribuídas através da rede juntamente com o restante do cenário (comum para todo o *cluster*). Essa comunicação inicial poderia ser implementada, por exemplo, utilizando bibliotecas de comunicação padrão MPI. Porém, uma estratégia mais simples seria a construção de arquivos de entrada personalizados para cada processo, distribuídos através de um sistema de arquivos de rede, como por exemplo, o NFS (*Network File System*), utilizado em sistemas UNIX. [12].

No final da simulação dos raios, cada processo separado poderia enviar seus resultados através da rede utilizando MPI, ou disponibilizando-os na forma de arquivos locais compartilhados através do NFS. O resultado do processamento seria um relatório de todos os raios lançados ao longo da história ou apenas dos raios que atingiram certas regiões pré-estabelecidas do cenário. Particularmente a relação de todos os raios lançados na história poderia constituir uma imensa quantidade de dados, cujo armazenamento em memória principal seria inviável. Assim, se for indesejável a gravação dos resultados em arquivos locais, será necessário despachá-los continuamente através da rede (por exemplo, com o auxílio da MPI), à medida que a simulação transcorre. A captação e unificação dos resultados gerados pelos processos do *cluster* consiste na etapa de pós-processamento (Fig. 1).

V. CONCLUSÕES

A utilização de processamento paralelo em conjunto com a técnica de traçado de raios (*ray tracing*) possibilitará uma maior aplicabilidade e eficiência desta última.

Percebe-se que a técnica SBR, ou Força Bruta se mostra naturalmente paralelizável, ao contrário da técnica IT. Isto possibilita uma implementação mais lógica e robusta do

ambiente paralelo, uma vez que o ambiente de análise facilita a quebra de uma tarefa maior em uma série de tarefas menores.

Aliando o processamento paralelo às técnicas de aceleração do particionamento do cenário, haverá um ganho substancial de performance na execução da técnica de traçado de raios, abrindo a possibilidade de análise de outros ambientes *wireless*.

Outra forma de melhoria de desempenho, também trabalhada no Grupo, é através do pré-processamento da base de dados. Uma vez que as relações de visibilidade entre as paredes internas e dos corredores de um prédio, por exemplo, não se alteram com a mudança de posição da estação transmissora ou receptora, tais dados podem ser pré-processados e armazenados para uso futuro, de tal forma que não seja necessário o processamento dos dados de todo o ambiente. Determinando-se as relações de visibilidade para um dado ambiente (*indoor* ou *outdoor*) e se realizando um pré-processamento destes dados, o tempo gasto na simulação de todo o sistema será sensivelmente menor.

REFERÊNCIAS

- [1] Y. Foster, K. Kennedy, J. Dongarra, G. Fox. Sourcebook of Parallel Computing, Morgan Kaufmman Pub, 2002.
- [2] W. Gropp, E. Lusk, A. Skjellum, R. Thakur. Using MPI : Portable Parallel Programming with the Message Passing Interface (Scientific and Engineering Computation), 2nd Edition, MIT Press, 1999.
- [3] W. Liu et al. "Parallel Simulation Environment for Mobile Wireless Networks". 1996 Winter Simulation Conference Proceedings (WSC '96), 1996.
- [4] G. Wölfle, R. Hoppe, F. M. Landstorfer. "A Fast and Enhanced Ray Optical Propagation Model for *Indoor* and Urban Scenarios, Based on an Intelligent Preprocessing of the Database". 10th IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, Setembro, 1999.
- [5] R. Hoppe, G. Wölfle, F. M. Landstorfer. "Accelerated Ray Optical Propagation Modeling for the Planning of Wireless Communication Networks", Nec Research Index (<http://citeseer.nj.nec.com/433917.html>), 2004.
- [6] M. J. de Sousa, L. V. de Souza, C. de S. Sales, R. da Silva, J. C. Chamma, C. R. L. Francês e J. C. W. A. Costa. "Otimização de Grades de Bragg em Fibra Usando Processamento Paralelo e Algoritmo Genético", Simpósio Brasileiro de Telecomunicações – SBrT, Agosto, 2003.
- [7] R. M. S. Oliveira, R. O. Santos and C. L. S. S. Sobrinho "Electromagnetic Scattering Analysis in Indoor and Outdoor Environments by Applying FDTD Method", IMOC 2003, Foz do Iguaçu, Agosto, 2003.
- [8] M. F. Catedra and J. Pérez-Arriaga. Cell Planning for Wireless Communications, Artech House –Mobile Communications Series, 1999.
- [9] D. N. Schettino e F. J. S. Moreira, "Um Algoritmo Eficiente para o Traçado de Raios na Predição de Cobertura Radioelétrica em Ambientes Urbanos", X Simpósio Brasileiro de Microondas e Optoeletrônica, Recife, PE, Brasil, , pp. 196-200, Agosto 2002.
- [10] G. Wölfle, R. Hoppe, T. Rautiainen, "Verifying path loss and delay spread predictions of a 3D ray tracing propagation model in urban environment", Vehicular Technology Conference, 2002. Proceedings. VTC 2002-Fall. 2002 IEEE 56th, vol.4, pp. 24-28 Sept. 2002.
- [11] R. A. Tenenbaum, J. G. Slama, M. L. Ballesteros, "Numerical simulation of room acoustics: a new approach for source modelling", Proceedings 14th International Congress on Acoustics, Beijing, 1992.
- [12] Tanenbaum, A.S. Computer Networks, 4a Ed., Prentice Hall, 2002.